Real Time Intelligent Target Detection and Analysis with Machine Vision*

Ayanna Howard, Curtis Padgett, Kenneth Brown
Jet Propulsion Laboratory, California Institute of Technology
4800 Oak Grove Drive, Pasadena, CA 91109-8099

ABSTRACT

This paper presents an algorithm for detecting a specified set of target objects embedded in visual imagery for an Automatic Target Recognition (ATR) application. ATR involves processing images for detecting, classifying, and tracking targets embedded in a background scene. We address the problem of discriminating between targets and non-target objects located within a cluttered environment by evaluating 40x40 image blocks belonging to a segmented image scene. Using directed principal component analysis, the data dimensionality of an image block is first reduced and then clustered into one of **n** classes based on a minimum distance to a set of **n** cluster prototypes. Following clustering, each image pattern is fed into an associated trained neural network for classification. A detailed description of our algorithm will be given in this paper. Evaluation of the overall algorithm demonstrates that our detection rates approach 96% with a false positive rate of less than 0.03%.

KEYWORDS: Target Detection, Neural Networks, Classification

INTRODUCTION

The steps required for successful implementation of an Automatic Target Recognition (ATR) task involves automatic detection, classification, and tracking of a target located, or camouflaged, in an image scene. As such, an ATR system must be invariant toward all vantage point differences. This includes illumination changes, image shadowing, perspective distortion, and target occlusion. For aerial ATR applications, real world imagery, using on-line aerial images, will be affected by climate, season, weather, and time of day variations. An aerial image is also subject to geometric changes, such as position, orientation, and scale disparities.

Normally, the target recognition process is highly dependent on apriori knowledge. Most systems are only capable of recognizing a pre-specified number of targets and are limited in their

* Presented at the Third International Symposium on Intelligent Automation and Control, World Automation Congress, 11-16 June 2000.

performance on novel (interpolated) data. In addition, many ATR systems are encoded with predetermined tolerances resulting in a tendency to be very sensitive to scale and orientation changes. MODALS [7], a 3-D multiple object detection and location system, utilizes a neural network to simultaneously segment, detect, locate, and identify multiple targets. Although MODALS is able to provide robust detection, high classification, and a low false alarm rate, it is not rotation or scale invariant. SAHTIRN [1] performs automatic target recognition through a three-stage process using an edge detector, a multi-layer feedfoward clustering neural network and a neural network classifier. SAHTIRN is able to successfully classify objects with varying scale and orientation parameters, but is not robust when faced with changes in lighting conditions. Greenberg and Guterman [3] use neural networks to address the issue of target classification, but assume target locations have previously been identified in a separate ATR-detection process.

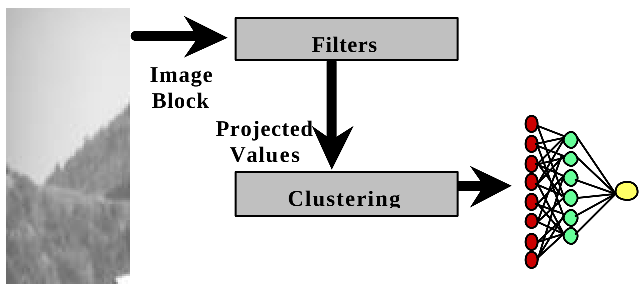

Figure 1: The data processing path for each image block extracted from video input. The image block is projected onto a set of filters, associated with a particular cluster, and then classified with the associated neural network.

To address these limitations, our research objective is to develop a novel technique that autonomously detects, in real time, all target objects embedded in a background image scene. The evaluation of these algorithms is based on two sets of data – real-world embedded target imagery and a dataset constructed from inserting target images into real background scenes acquired from video input. In real time, the data dimensionality of a scene will be reduced using an optimal set of templates and targets spatially located in the scene will be identified with a neural network classifier. Figure 1 provides an overview of the approach used for detecting a known set of targets in a background image scene. The rest of this paper describes the autonomous target detection methodology in detail.

TECHNIQUE

Background and Target Data Set

The image scenes used in this research effort are acquired from video camera from the JPL in Pasadena site. We segment these background images into 40x40 image blocks for input into our algorithm (Fig. 2:Top). Target objects (Fig. 2:Middle) are modeled from an actual cruise missle and represent various scale and rotation perspectives of the missle. These synthetic target objects are used for training the algorithm and are embedded into a background image block such that:

$$I^T = \begin{cases} b_{xy} : t_{x,y} = 0 \\ 0.6t_{x,y} + 0.1(t_{x-1,y} + t_{x+1,y} + t_{x,y-1} + t_{x,y+1}) : t_{x,y} <> 0 \end{cases}$$

where t_{xy} is the pixel intensity value of the target image at (x,y), b_{xy} is the pixel value of the background image at (x,y), and I^T is the embedded target-background image block. Examples of embedded target-background image blocks are shown in Figure 2:Bottom.

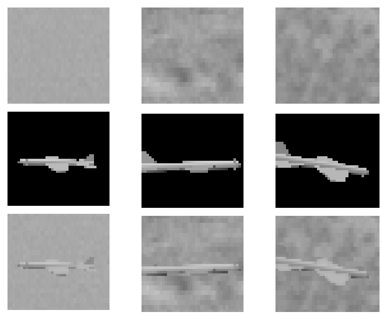

Figure 2. Top Row: Background Images; Middle Row: Target Objects; Bottom Row: Background Images with Embedded Targets

Once we extract our background and embedded target data set, we perform a preprocessing step in order to account for time-of-day lighting variations in the image set. We subtract the average image block intensity value from each pixel such that:

$$I^c = I - \frac{\sum i_{xy}}{N}$$

where i_{xy} is the image intensity of pixel (x,y) in image block I, N is the size of image block I (40x40), and I^c is the corrected image block to be used in our algorithm.

Using this data set, we train an algorithm capable of intelligently detecting a target embedded within a background image scene. The next section describes our approach for the development of such an algorithm.

Algorithm Description

Given a set of targets **T**, the goal of the algorithm is to detect in real time, any target **t** Î **T** present in a 40x40 image block extracted from a background scene. After the data preprocessing step, we begin by projecting an image block onto a set of templates specifically designed to separate signatures derived from a target embedded in a background from other typical background image signatures. These projections, or patterns, are then clustered into one of **n** classes based on their distance to a set of **n** cluster prototypes. These cluster prototypes have previously been identified using a modified clustering algorithm based on prior sensed data.

Associated with each cluster is a trained neural network classifier. After clustering, the projected image pattern is fed through this associated trained neural network for detection.

In order to accomplish our target detection goal, prior knowledge must be derived through the following algorithmic preprocessing steps:

i. Derive a set of linear filters used to optimally separate targets embedded in a background image scene from other typical background images.
ii. Identify a set of cluster prototypes used to classify the projected image patterns.
iii. Train a set of *expert* neural network classifiers for each cluster which responds with 1 when fed embedded target-background image patterns and –1 otherwise.

Linear Filter Sets. The filtering step involves an orthogonal sub-space projection of each image block. It is used to optimally linearly separate the embedded target background images from those images not containing targets. This is a standard technique used to reduce the dimensionality of the image block [from 1600 (40x40) to 17 dimensions] while preserving as much of the signal as possible. The filters associated with a given prototype are derived from the distribution of a background image (***noise***) and the distribution of potential targets embedded in that background (***signal***). This can be optimally separated to maximize the *signal to noise* ratio between the two groups using directed principal components analysis (DPCA). To characterize the distribution for the background image, the covariance matrix, $\mathbf{R}_i$, is found for image blocks which do not contain targets. We characterize the mixed target-background image distribution instances by its covariance matrix, $\mathbf{S}_i$.

We are interested in finding a set of orthogonal basis vectors $\mathbf{W}_i$, that maximizes the expected signal to noise ratio of these two distributions defined by their respective image sets. The generalized eigenvector solution:

$$\mathbf{S}_i \mathbf{W}_i = \lambda \mathbf{R}_i \mathbf{W}_i$$

accomplishes this. The set of filters defined by $\mathbf{W}_i$ is the directed components used in our algorithm. They essentially steer the eigenvector solution away from dimensions of high noise variance in a linearly optimal fashion. Figure 3 shows a subset of the filter set.

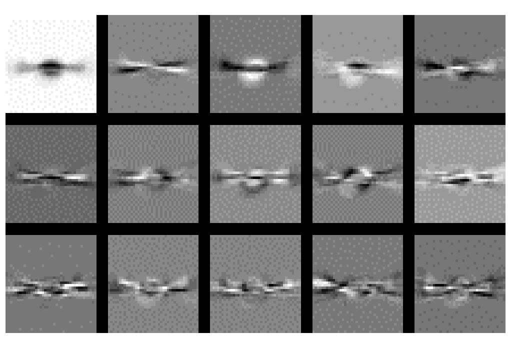

Figure 3. Top 15 directed principle components

Clustering. To effectively simplify the distribution of data classified by an expert neural network, we partition the incoming projected image patterns drawn from a known distribution of background and embedded target-background images into a number of predetermined groups by using the prototypes $\mathbf{P}_i$ of a clustering algorithm. The clustering algorithm is run on previously acquired data that reflects the distribution of the scene being analyzed.

The clustering algorithm employed is a modified version of a standard clustering technique outlined in Duda and Hart [2]. The standard algorithm uses a standard least squares criterion to minimize the distance between each of **n** randomly selected groups. We modify this clustering algorithm to more likely group embedded target-background patterns together while still discarding those background patterns which may have similar characteristics. This change consists of embedding a simple weighting term reflecting variance into the criterion calculation. The criterion minimized by the modified clustering algorithm thus becomes:

$$\text{cost} = \sum_i (1+w_i) \sum_j \| p_j - P_i \| \tag{1}$$

where

$$w = \frac{nt_i}{np_i}\left(1 - \frac{nt_i}{np_i}\right)$$

i is one of **n** clusters, p_j is a projected image pattern in that cluster, nt_i is the number of embedded target patterns in cluster i, and np_i is the number of background patterns in cluster i. The clustering algorithm iterates through each projected image pattern and determines if moving the pattern to another group reduces the overall cost. Patterns that are not like those already in the cluster will be weighted more in the cost of the clustering algorithm than those alike thus allowing the clustering algorithm to naturally link alike elements together. If movement reduces cost, the pattern is moved to the other group and the associated averages of each prototype cluster are recalculated. This continues until the moving of patterns no longer reduces the overall cost. The resultant cluster prototypes are then employed by our algorithm to segment the image scene. Figure 4 shows a background scene segmented with the derived clustering prototypes.

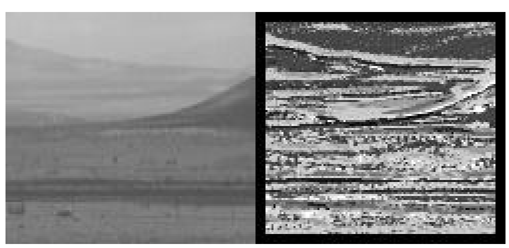

Figure 4. A Segmented Background Image Scene

Classification. The last step in our preprocessing algorithm involves classifying each projected image pattern belonging to cluster_i with a neural network [4]. The networks are trained with data drawn from the two distributions: background patterns $\mathbf{R}_i$ and embedded target-background patterns $\mathbf{S}_i$. The expert network for class **i** is required to respond with 1 for elements drawn from $\mathbf{S}_i$ and –1 from those drawn from $\mathbf{R}_i$. We use a simple feed forward network model employing 17 inputs and 10 sigmoidal hidden units trained with backpropogation to get the desired result. The output can then be thresholded to achieve the desired detection rate or false positive rate by examining the receiver operator curves.

Algorithm Implementation

After we implement the preprocessing steps, we perform the real-time intelligent target detection process. After subtracting out the mean, each image block is projected onto the linear filter set. The projected image pattern is then compared to the set of pre-computed cluster prototypes. Based on the Euclidean distance, the pattern is grouped with the closest prototype.

The trained neural network classifier is then utilized to evaluate whether or not the image pattern contains a target from **T**. The neural network for each cluster group takes as input the projected values of the image and outputs a value. Values above a threshold are considered images with targets and those below are assigned to background.

The effectiveness of the evaluation requires that the cluster prototypes generated and the image blocks used in training the classifier must be derived from scenery with roughly the same distributions as encountered in the operational test.

RESULTS

We evaluated the overall performance of the algorithm using the described target set and background images.

The background scenes consisted of over one million image blocks of which less than 5% were used in developing a set of training data. Testing data consisted of randomly drawn image blocks from the background scenes. Embedded target images were generated by randomly selecting images from the target set and mixing them with arbitrary background image blocks. A sub-sample of the training data (1000 examples each) was used to generate the covariance matrixes **R** and **S**. The generalized eigenvector solution **W** was then solved using Matlab. The training data was then projected onto the filter set and evaluated with the clustering technique to realize the cluster prototypes (**P**) used in step ? of the preprocessing algorithm.

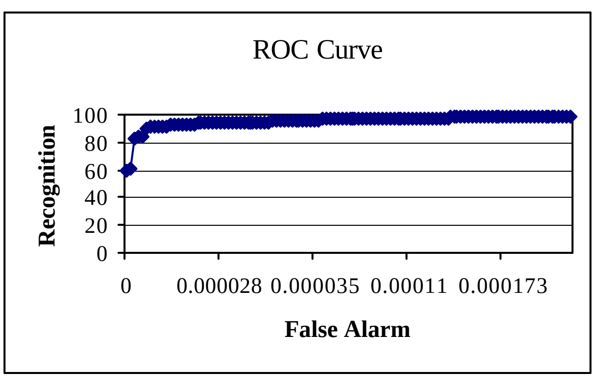

Figure 5: Detection vs False Positive Rate

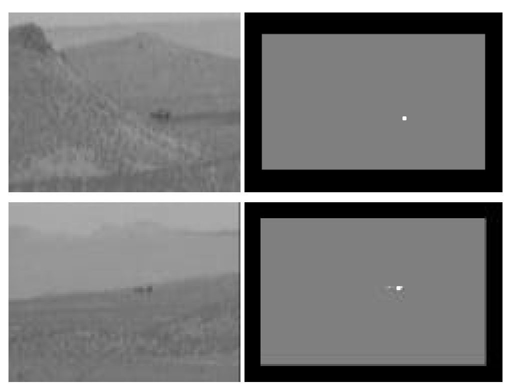

Figure 6: Detection Output

Training data for the neural network was again drawn from the set of training image blocks. In addition, a portion of the training data for the network was used to halt training (a hold out set) as described in Haykin [4]. Training of the networks used 80,000 examples, ½ target and ½ background images. The hold out set consisted of 40,000 examples not trained upon.

Our results, based on real-world target imagery and the embedded target dataset, give us a detection rate of 96% with a false positive rate of less than 0.03%. These results are constructed with 100,000 novel 40x40 image blocks. Figure 5 shows our Receiver Operator Curve (showing Detection vs. False Positive Rate) and Figure 6 shows an example of the detection output.

CONCLUSION

A novel detection algorithm and our evaluation methodology are described here. The detection algorithm was shown to perform detection at a rate of 96% with false positives less than 0.03% on a set of targets mixed with background images.

ACKNOWLEDGEMENTS

The research described in this paper was carried out by the Jet Propulsion Laboratory, California Institute of Technology, and was sponsored by the Ballistic Missile Defense Organization through an agreement with the National Aeronautics and Space Administration.

Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not constitute or imply its endorsement by the United States Government or the Jet Propulsion Laboratory, California Institute of Technology.

REFERENCES

1. C.E. Daniell, D.J. Kemsley, W.P. Lincoln, W.A. Tackett, G.A. Baraghimian, “Artificial neural networks for automatic target recognition”, Optical Engineering, Vol. 31(12), pp. 2521-2530, Dec. 1992.
2. H. Duda, M. Hart, *Pattern Classification and Scene Analysis*. John Wiley & Sons. New York. 1973.
3. S. Greenberg, J. Guterman, “Neural-network classifiers for automatic real world aerial image recognition”, Applied Optics, Vol. 35, No. 23, pp. 4598-4608, Aug. 1996.
4. S. Haykin, *Neural Networks: A Comprehensive Foundation*. Macmillan College Publishing Company. New York. 1994.
5. A. Howard, C. Padgett, C.Liebe, "A Multi-Stage Neural Network for Automatic Target Detection" in proceedings 1998 IEEE World Congress on Comp. Intelligence, May 98.
6. S.K. Rogers, J.M. Colombi, C.E. Martin, J.C. Gainey, K.H. Fielding, T.J. Burns, D.W. Ruck, M. Kabrisky, M. Oxley, “Neural Networks for Automatic Target Recognition”, Neural Networks, Vol. 8, No. 7/8, pp. 1153-1184, 1995.
7. W.A. Thoet, T.G. Rainey, D.W. Brettle, L.A. Slutz, F.S. Weingard, “ANVIL neural network program for three-dimensional automatic target recognition”, Optical Engineering, Vol. 31, No. 12, pp. 2532-2539, Dec. 1992.